

## 第八章-色域映射

在之前的色调映射章节中提到：在色调映射环节，为了便于操作，且不使图像颜色产生巨大失真，色调映射算法通常会仅处理图像亮度信息，将 HDR 图像亮度映射到 SDR 图像亮度域中，通过原 HDR 图像的颜色信息，恢复并重建 SDR 图像的颜色信息。由于前面的主题是色调映射，因此颜色转换相关技术没有深入介绍。但颜色转换或色域映射问题（Color Transfer or Gamut Mapping），也是 HDR 的重要环节。本章将介绍 HDR 中颜色转换(或色域映射)技术。本章分为两个部分，第一部分介绍色域映射的定义以及相关背景知识；第二部分将介绍代表性的色域映射算法，特别对 ITU 中相关标准进行浅析。

### 8.1 色域映射概述

在日常生活中，人们会在多种场景和设备上观看图像或视频。由于显示设备的多样性，自然会引发一个问题：给定一幅图像，如何能够保证人们在这些不同的显示设备上看到的图像是一致的？更进一步，如何使得不同显示设备上看到的图像，与这幅图像的原场景是一致的？

由于不同显示设备原理、设计、颜色标准不尽相同，想要实现这一目标是非常困难的。如果希望不同的人在不同的显示设备上看到的图像一致，首先需要了解哪些原因造成不一致。原因可以大致分为三类：

1. 不同人视觉特性不同。不同的人，对于进入眼睛的光线的响应是存在轻微不同的，因此，即使在同一场景下观察同一图像，不同的人对于图像的感知（例如颜色，对比度，饱和度等）都是不同的。
2. 观测场景不同。同样的图像，在不同的场景下观测，同样观测者感受也是不同。
3. 显示方式不同。对于一幅数码相机拍摄的图像，将存储的图像文件使用不同的显示设备显示，人们观察到的图像也是不同的。

色域，来用于描述不同显示设备，或者不同标准所使用或规定的颜色范围。而在两个不同色域之间构建的映射，称为色域映射。由于视觉特性以及场景原因带来的差异难以建模并且处理，因此色域映射问题通常是指显示设备的显示方式不同产生的色域差异。一般来说，可以通过改变像素值将两个色域中均含有的颜色进行映射，以实现颜色的一致。但当两个色域范围有差别，也就是说一个色域中包含了另一个色域没有的颜色时，色域映射就比较复杂，通常的做法是选择最接近于原图像颜色的颜色，进行显示。

在 CIE(2004d) [2]标准中，给出色域映射及其相关内容的官方定义：

**色域映射：**一种将待显示媒介的颜色用原始图像或媒介中的颜色表示的方法

**图像：**一种包含了图形与形象的二维视觉刺激。重新生成的图像可以通过某些特征与原始图像进行比较。

**颜色媒介：**能够显示或者获取颜色的媒介物。例如 CRT 显示器，数码相机。

**色域：**在给定的观测条件下，一个给定的颜色媒介能够显示或获取的颜色范围。

由此可以明确，**色域映射是将两种不同的色域联系起来的核⼼方法**。色域映射的分类主要包括四大类：独立（两个色域之间没有交集），交叉（两个色域相交，但不包含），缩小（目标色域小于原色域），扩张（目标色域大于原色域），具体见图 1。其中 S 表示原色域，D 表示目标色域。

前面部分介绍了色域映射的基础，后续部分中将把注意力转向 HDR 中是如何进行

色域映射的。更多关于传统色域映射的内容与算法，可参阅文献 [1]，该书包含了关于传统色域映射（即不同显示设备之间）的丰富内容。

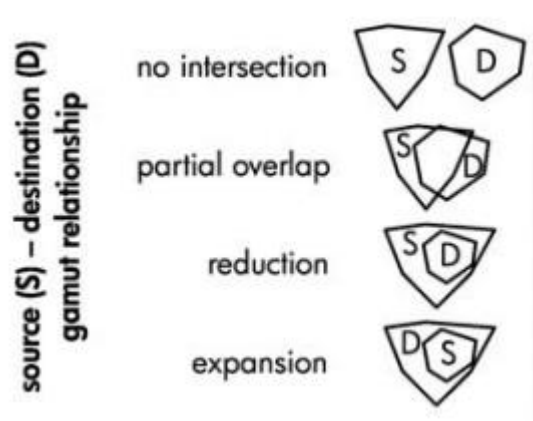


图 8-1 色域映射的分类[1]

## 8.2 HDR 中的色域映射标准与算法

在 HDR 技术中，色域映射的目的有所不同。传统的色域映射方法，是为了解决不同显示设备之间色域不同的问题，但是归根结底，传统显示设备的色域都在 BT.709 [3] 标准范围内。而在 HDR 中，由于 HDR 图像的亮度与颜色是与场景亮度与颜色一致的，因此理论上来说，HDR 图像显示在不同的 HDR 显示设备上时，颜色与亮度也不会存在差异，因此 HDR 的色域映射方法并不是用于解决不同显示设备的颜色不一致问题的，而是主要用于色调映射过程中的颜色处理。由于色调映射过程中，HDR 图像映射成了 SDR 图像，因此，图像的亮度与颜色都发生了较大的改变。而对于颜色来说，由于 HDR 图像的色域标准遵循 BT.2020 [4] 标准，而 SDR 图像遵循 BT.709 标准，因此，本质上来说，HDR 中的色域映射问题，实际上就是 BT.2020 色域与 BT.709 色域之间的转换问题。下面，我们将从 ITU 标准入手，介绍 HDR 的色域映射标准与相关算法。

### 8.2.1 BT. 709 与 BT. 2020 颜色空间

在 1931 年，CIE 根据相关实验，测定了人眼能够观察到的颜色范围，由此确定了“CIE 1931 标准观察者标准”，也被称为 CIE-XYZ 表示法。而在图像与视频处理领域，显示设备所能显示的颜色，远少于自然界中的颜色范围，由于技术的限制以及显示设备的制作原理的限制，现在的显示设备无法显示出自然界中存在的所有颜色，而只能选择其中的一部分颜色进行显示。

在 HDR 技术中，通常使用的颜色标准是 ITU-R BT. 2020 标准。BT. 709 是 HDTV 的标准，也是目前运用最为广泛的颜色标准。BT. 2020 是 UHDTV 的颜色标准，也是 HDR 技术所支持的颜色标准。HDR 中的色域映射，就发生在这两个色域之间。

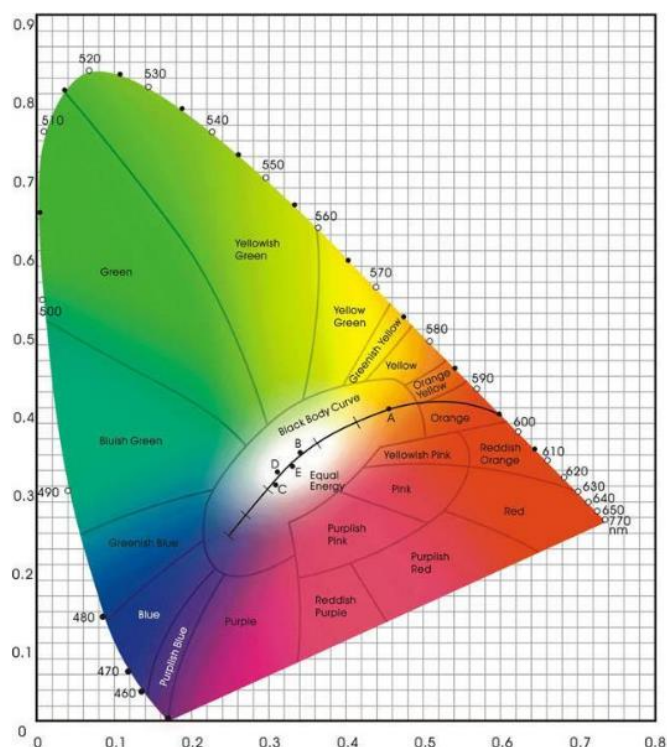


图 8-2 CIE-XYZ 1931

## BT. 709 色域

BT. 709 标准中定义了 SDR 图像的编码方式。具体分为  $R'G'B'$  和  $Y'C_bC_r$  的两种编码模式，每种模式的各通道都有 8 或 10 比特两种类型，在 8 位编码中， $R'G'B'$  通道和  $Y'$  通道的标称范围是 16~235， $C_b$  和  $C_r$  通道的标称范围是 16~240，中性值为 128。因此，在  $R'G'B'$  中，黑色的值是 (16, 16, 16)，白色的值是 (235, 235, 235)；在  $Y'C_bC_r$  中黑色的值是 (16, 128, 128)，白色的值是 (235, 128, 128)。标准 RGB 空间的 0 和 255 仍保留作为定时参考，可能不包含颜色数据。BT. 709 色域的范围见图 3。

BT. 709 标准同时支持模拟信号与数字信号，因此在 BT. 709 标准内部，存在两个重要的转换： $R'G'B'$  编码到  $Y'C_bC_r$  编码的转换，以及模拟信号到数字信号的转换。

$R'G'B'$  编码下的模拟信号到  $Y'C_bC_r$  编码下的模拟信号之间的转换为：

$$\begin{aligned} E_Y &= 0.2126E_R + 0.7152E_G + 0.0722E_B \\ E_{CB} &= \frac{-0.2126E_R - 0.7152E_G + 0.9278E_B}{1.8556} \\ E_{CR} &= \frac{0.7874E_R - 0.7152E_G - 0.0722E_B}{1.5748} \end{aligned} \quad (1)$$

其中  $E_Y, E_{CB}, E_{CR}$  分别表示亮度信号和两种色差信号， $E_R, E_G, E_B$  分别表示  $R', G', B'$  三个通道的模拟信号值。

两种编码方式的数字图像的量化方法是：

$$\begin{aligned}
D_{R,G,B} &= INT[(219E_{R,G,B} + 16) \cdot 2^{n-8}] \\
D_Y &= INT[(219E_Y + 16) \cdot 2^{n-8}] \\
D_{CB} &= INT[(224E_{CB} + 128) \cdot 2^{n-8}] \\
D_{CR} &= INT[(224E_{CR} + 128) \cdot 2^{n-8}]
\end{aligned} \tag{2}$$

其中  $n$  表示量化信号的比特长度，INT 作为将小数化整的四舍五入功能。因此可以得出  $R'G'B'$  编码下的数字信号到  $Y'C_bC_r$  编码下的数字信号之间的转换方法：

$$\begin{aligned}
D_Y &= INT[0.2126D_R + 0.7152D_G + 0.0722D_B] \\
D_{CB} &= INT\left[\left(-\frac{0.2126}{1.8556}D_R - \frac{0.7152}{1.8556}D_G + \frac{0.9278}{1.8556}D_B\right) \cdot \frac{224}{219} + 2^{n-1}\right] \\
D_{CR} &= INT\left[\left(\frac{0.7874}{1.5748}D_R - \frac{0.7152}{1.5748}D_G - \frac{0.0722}{1.5748}D_B\right) \cdot \frac{224}{219} + 2^{n-1}\right]
\end{aligned} \tag{3}$$

## BT. 2020 色域

ITU-R BT. 2020 定义了 UHDTV 的各个方面，比如显示分辨率、帧速率、色度子采样、比特深度和色彩空间，它是由 ITU 在 2012 年发布的。相比于 BT. 709，BT. 2020 提供了更加广泛的颜色范围，BT. 2020 也是 HDR 技术的颜色空间。

BT. 2020 标准的重要性是毋庸置疑的，正如 BT. 709 对于高清视频传输与高清显示设备制造方面起到的引导性作用，BT. 2020 标准同样也深刻地影响着消费领域超高清显示设备的设计与制造，例如目前绝大部分的 4K 平板电视的物理分辨率都是采用 BT. 2020 标准的  $3840 \times 2160$ ，但 BT. 2020 标准绝不仅仅在分辨率方面有所提升，在色彩和刷新频率和信号格式与分析等方面也进行了相关的规定。本文仅关注 BT. 2020 标准中关于颜色空间的定义。

相对于 BT. 709 标准，BT. 2020 标准大幅度提升了视频信号的性能规范，色彩深度方面提高至 10bit 或 12bit，很大程度上增强了整个影像的色彩层次，色域的广度也大大增强，两种标准的色域范围对比如图 3 所示。

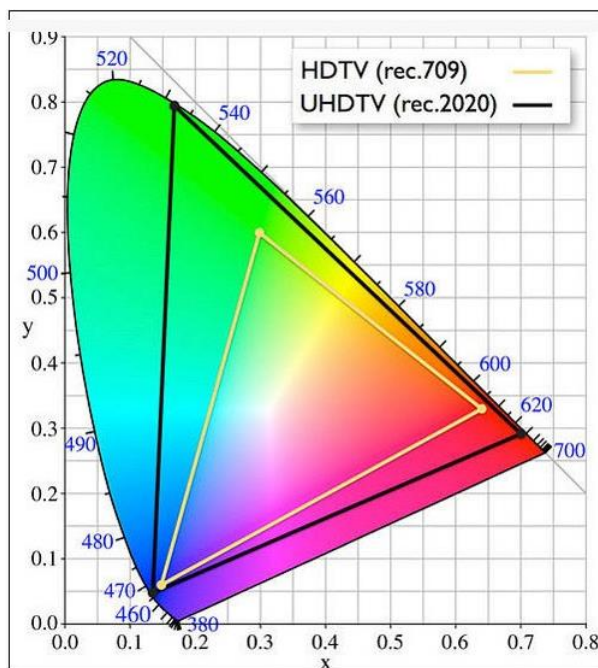


图 8-3：BT. 709 与 BT. 2020 对比

与 BT. 709 一致，BT. 2020 也支持了  $R'G'B'$  和  $Y'C_bC_r$  两种编码模式，以及支持模拟与数字两种信号。同样的，BT. 2020 标准内部也存在两个重要的转换：两种编码模式之间的转换以及模拟信号到数字信号的转换。

$R'G'B'$  编码下的模拟信号到  $Y'C_bC_r$  编码下的模拟信号之间的转换：

$$\begin{aligned} E_Y &= 0.2627E_R + 0.6780E_G + 0.0593E_B \\ E_{CB} &= \frac{-0.2627E_R - 0.6780E_G + 0.9407E_B}{1.8814} \\ E_{CR} &= \frac{0.7373E_R - 0.6780E_G - 0.0593E_B}{1.4746} \end{aligned} \quad (4)$$

$R'G'B'$  编码下的数字信号到  $Y'C_bC_r$  编码下的数字信号之间的转换：

$$\begin{aligned} D_Y &= INT[0.2627D_R + 0.6780D_G + 0.0593D_B] \\ D_{CB} &= INT\left[\left(-\frac{0.2627}{1.8814}D_R - \frac{0.6780}{1.8814}D_G + \frac{0.9407}{1.8814}D_B\right) \cdot \frac{224}{219} + 2^{n-1}\right] \\ D_{CR} &= INT\left[\left(\frac{0.7373}{1.4746}D_R - \frac{0.6780}{1.4746}D_G - \frac{0.0593}{1.4746}D_B\right) \cdot \frac{224}{219} + 2^{n-1}\right] \end{aligned} \quad (5)$$

## BT. 2020 色域到 BT. 709 色域的转换

HDR 中的色域转换问题，特别是色调映射之后的色域映射，主要是 ITU-R BT. 2020 色域到 ITU-R BT. 709 色域的颜色转换。理想情况下转换应该满足以下要求：1. BT. 709 色域内的颜色应该保持不变；2. 转换方法适用于 BT. 2020 和 BT. 709 之间的多次转换；3. 能够感知到的色调变化必须尽可能小；4. 空间细节没有重大损失；5. 不会在颜色上引入可见的不连续性；6. 映射方法在数学上是可定义的。

需要注意的是，没有可以同时满足上述这些要求的通用色域映射方法。在从更广泛的色域转换到更小的色域时，BT. 709 色域外的颜色出现改变是不可避免的。不同的转换方式会在不同要求之间进行妥协，具体的做法可能因应用场景而异。下面介绍一种 ITU-R BT. 2407 [5] 标准中介绍的基于线性矩阵的从 BT. 2020 到 BT. 709 的简单变换。

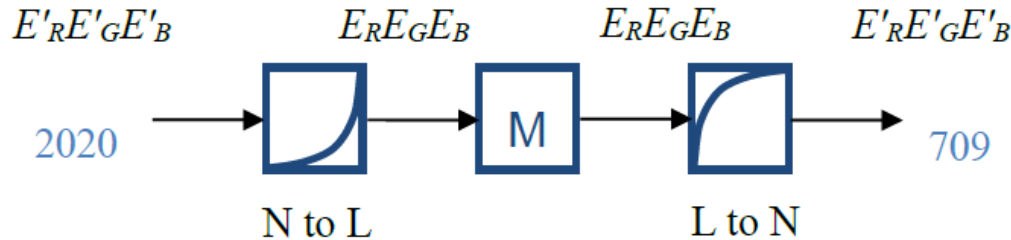


图 8-4 基于线性矩阵变换从 BT. 2020 到 BT. 709 的颜色转换

### ITU-R BT. 2407 中的简单色域映射方式

图 4 是从 BT. 2020 到 BT. 709 的基于线性矩阵变换的颜色转换的框图。除了输出信号被严格限制之外，这是 ITU-R BT. 2087 [6] 标准中对从 BT. 709 转换到 BT. 2020 所规定的操作的逆操作。这种方法在条件限制的情况下是最直接简单的。

这种方法首先使用 ITU-R BT. 2087 中规定的两个非线性到线性的传递函数（一种是电光转换功能函数（EOTF），一种是逆光电转移函数（OETF））之一将归一化的非线性 RGB 信号转换为归一化的线性 RGB 信号。接着通过矩阵运算，将 BT. 2020 颜色空间的 RGB 信号转换到 BT. 709 颜色空间的 RGB 信号。最后再通过使用 ITU-R BT. 2087 建议书规定的两种线非线性传递函数（一种是逆 EOTF 方法，另一种是使用 OETF）之一将线性 RGB 信号转换为非线性 RGB 信号。从归一化线性 RGB 信号（ $E_R E_G E_B$ ）到归一化非线性 RGB 信号（ $E'_R E'_G E'_B$ ）的转换。

该方法具有非常理想的特性，即使在 BT. 2020 和 BT. 709 之间进行多次转换之后，它也不会改变 BT. 709 色域内的颜色。然而，BT. 709 色域之外的色彩是硬限幅的，即小于零或大于 1 的 RGB 信号（ $E_R E_G E_B$ ）会被限幅为零或 1，这会导致明显的色调偏移和空间细节的损失。虽然大多数内容看起来都很好，但色调和空间细节方面还是会有所损失。

### ITU-R BT. 2407 中规定的色域映射算法原则

为了提升简单色域映射算法的结果，BT. 2407 提供了一套色域映射算法的设计流程与设计原则。图 5 显示了从 BT. 2020 色域到 BT. 709 色域的算法抽象框图。首先，选定一个映射颜色空间，然后在算法的第一步，将输入的符合 BT. 2020 标准的 RGB 表示的图像转换为映射颜色空间的形式。接着，使用色域映射算法，将映射颜色空间表示的颜色转换到 BT. 709 色域内。最后，再将颜色空间转换为 RGB 表示。到此，色域映射流程结束。

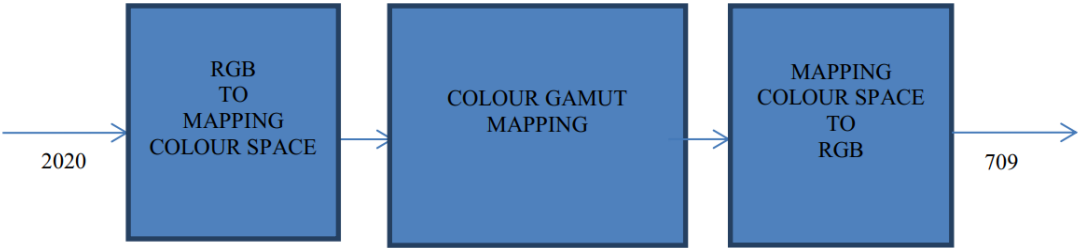


图 8-5 BT. 2020 色域到 BT. 709 色域转换算法流程图

其中，映射颜色空间的选取是算法的一个重要方面，可供选择的映射颜色空间包括：



1. CIE xyY
2. CIE u' v' Y
3. CIE  $L^*a^*b^*$
4. 简化版 Lab
5. 基于 CIECAM02 的 Uniform Color Space

BT. 2020 色域与 BT. 709 色域的 RGB 值( $E_R, E_G, E_B$ )可以通过线性转换与投影, 转换到 CIE xyY 以及 CIE u' v' Y 的颜色空间中。CIE xyY 表示法中的 xy 平面代表传统的色度 (Chromaticity), CIE u' v' Y 表示法中的 u' v' 平面也表示色度, 但是当 Y 一定时, 相对于 xy 平面来说, u' v' 表示法下的色度更加的一致 (Uniform)。

当使用 CIE xyY 以及 CIE u' v' Y 作为映射颜色空间时, 可以通过线性变换来简化色域映射算法。但是, 需要注意的是, 简化后的某条映射路径 (即从 BT. 2020 色域的某一个颜色映射到 BT. 709 色域的某一个颜色) 是直线时, 颜色的色度, 色相和明度会同时出现改变。

CIE  $L^*a^*b^*$  颜色空间相对于 CIE xyY 以及 CIE u' v' Y 颜色空间说, 提供了更加优秀的视觉统一性。而简化版 Lab 是 CIE  $L^*a^*b^*$  的简化版。但是将 BT. 2020 与 BT. 709 的 RGB 值转换到这两种表示法时, 相对于 CIE xyY 以及 CIE u' v' Y 表示法来说, 需要使用更加复杂与非线性的操作。这两种颜色空间的优势在于由于将明度纳入了表示法的设计当中, 因此他们在感知上提供了更好的一致性。当使用这两种表示法时, 线性映射路径能够保留色度, 色相以及明度中的一个或两个属性。通过这些映射路径, BT. 2020 的颜色就能够映射为 BT. 709 的颜色。这种颜色的修改过程会使用到线性或非线性的方程。

在从 BT. 2020 到 BT. 709 的转换过程中, 一些色域映射算法将 BT. 709 色域内的颜色完全保留, 不加改变, 所做的颜色映射只针对那些在 BT. 709 之外, BT. 2020 之内的颜色。但是另外一下算法对 BT. 709 内部的颜色也作了一些改变, 特别是对处于 BT. 709 色域边缘的颜色。而当颜色在 BT. 709 色域与 BT. 2020 色域之间不断循环转换时, 这种对 BT. 709 色域内部颜色也做出改变的算法就会产生非常严重的人工痕迹。

从上述内容可以发现, BT. 2407 标准提供了一种简单的色域映射的算法, 同时, 也给出了一套设计流程。不同的设计者可以通过选择不同的色域表示法, 设计不同的色域映射算法, 来实现 BT. 2020 色域与 BT. 709 色域之间的转换。

## ITU-R BT. 2407 中的其他色域映射方法

BT.2407 中提供了 6 种较新的, 效果更好的色域映射方法, 本文中只给出这些色域映射方法的算法流程图或算法思路以供参考。算法的具体实现方法以及具体细节, 可以参见 [5]。

### 算法 1

这种算法在三个步骤中, 分别对源 HDR 图像的色相, 亮度与色度进行映射。算法框图如下:



图 8-6 算法 1 框图

### 算法 2

这种算法对图像的色相，亮度与色度同时进行映射。同时，在使用算法时，需要将图像使用 CIECAM02 表示法进行表示。

### 算法 3

算法 3 可以看做算法 1 的简化版本。它的算法流程图与图 6 中算法 1 的流程图一致，两者的区别是算法 3 在 CIE LAB 颜色空间中进行，同时利用简单的线性化操作来简化计算量，从而实现运算效率的加强。

### 算法 4

算法 4 提供了一种自适应的色域映射算法，这种算法的目的是尽可能保留原图的整体感知。具体流程图见图 4。

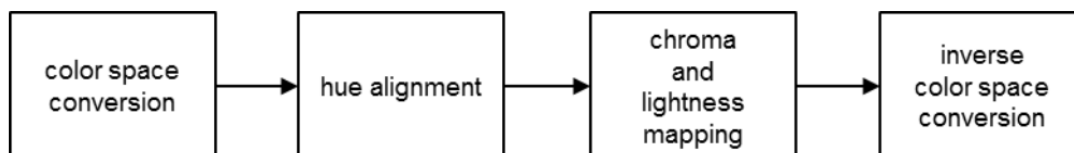


图 8-7 算法 4 流程图

### 算法 5

算法 5 是一种保留亮度信息的色域转换。通过改变图像的颜色空间，将图像的色度信息与亮度信息分开，从而仅仅处理图像的色度信息

### 算法 6

算法 6 的设计思路是在考虑显示设备色域的基础上，实现 BT.2020 空间中的颜色到 BT.709 空间中的颜色的比色法转换

## BT. 709 色域到 ITU-R BT. 2020 色域的转换

HDR 色域转换的另外一个角度就是从 BT. 709 色域到 BT. 2020 色域的转换。在将 SDR 内容适配到 HDR 显示设备上播放的过程中，通常需要将 BT. 709 色域中的颜色转换到 BT. 2020 色域中。

BT. 709 到 BT. 2020 的转换有一个重要的限制，即 BT. 709 标准下的颜色，在 BT. 2020 标准下要是同样的颜色。换句话说，如果这个颜色是两种色域中都存在的，那么这个颜色在转换过程中应当保持不变；而如果这个颜色是 BT. 2020 中独有的，这就需要进行颜色转换。BT. 709 到 BT. 2020 的转换可以大致分为三个主要步骤：第一个步骤是逆量化，得到 BT. 709 标准下的模拟信号值  $E_R, E_G, E_B$ 。第二个是色域转换，将 BT. 709 下的模拟信号值，转换为 BT. 2020 标准下的模拟信号值。最后一个步骤是量化，得到 BT. 2020 标准下的数字信号。选择在模拟信号域进行色域转换而不是直接在数字信号域进行色域转换的原因是为了保留更多的信息与细节。由于量化过程会存在信息的丢弃，并且 BT. 709 的色域比 BT. 2020 小，如果直接用数字信号进行转换，会使得 BT. 2020 色域下的图像效果变差。因此，通过逆量化过程，恢复图像的相关细节，然后进行模拟信号的色域转换，就可以解决这个问题。

BT. 709 色域到 BT. 2020 色域的转换标准遵循 BT. 2087 标准 [6]。标准中根据信号源的亮度是否恒定，将转换方式分为以下两种：

#### 1. 非恒定亮度信号的转换



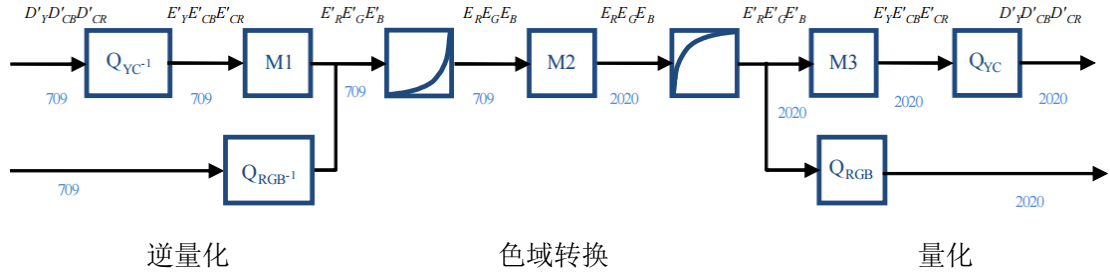


图 8-8. BT. 709 到 BT. 2020 在非恒定亮度信号格式下的转换过程

上图是非恒定亮度信号的转换流程图。由于 BT. 709 与 BT. 2020 均支持两种编码方式： $R'G'B'$ 和 $Y'CbCr$ ，因此逆量化过程的输入与量化过程的输出均有两条路径。根据需求的不同，可以选择不同的输入输出模式。

每一个区块的具体实现过程以及相关函数定义，可以参考 [6]。

## 2. 恒定亮度信号转换

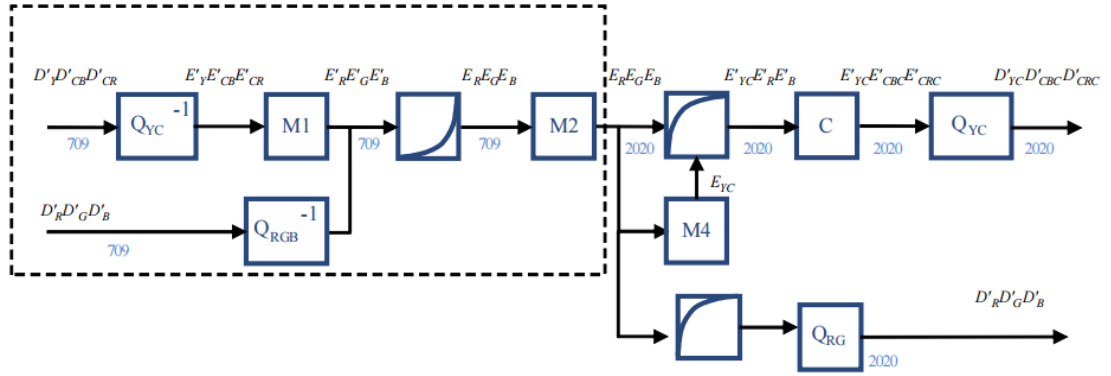


图 8-9. BT. 709 到 BT. 2020 在恒定亮度信号格式下的转换过程

上图是恒定亮度信号的转换流程图。由于 BT. 709 与 BT. 2020 均支持两种编码方式： $R'G'B'$ 和 $Y'CbCr$ ，因此逆量化过程的输入与量化过程的输出均有两条路径。根据需求的不同，可以选择不同的输入输出模式。

每一个区块的具体实现过程以及相关函数定义，可以参考 [6]。

## 8.3 色域映射算法

由于 BT.2407 标准只是提供了标准，指导思想与设计原则，因此有部分色域映射算法的映射效果要优于标准中推荐的算法。下面，将介绍一种近年来新提出的，并且效果优秀的色域映射算法。

### 8.3.1 HDR 颜色精确重构色域映射框架

[14]这种色域映射框架的优势在于，它能够将 HDR 图像的颜色精确地映射到 SDR 图像的颜色范围内。同时，它的运算量与其他色域映射算法相比要小，也可以很好地与现有的色调映射（Tone mapping）算法兼容。这种算法选择的映射颜色空间是 CIE LCh 表示法。将颜色转换为了 CIE LCh 表示法后，将色度通道进行压缩，使得色域能够与输出色域一致。这样的好处是避免了色相以及明度的移动，从而避免了一些严重的人工痕迹。

这种算法的流程图如图 10 所示。算法的输入是 XYZ 表示法下的 HDR 图像  $I$ 。首先，图像的亮度通道  $I(L)$  被已有的色调映射算法压缩。色度通道  $I(C)$  通过下文将要介绍的色度压缩算法进行校正，以消除色调映射过程中产生的不需要的饱和度。最后，图像的色度与亮度信息  $I(C,L)$  通过色域截断步骤进行处理，以保证每个像素点的像素值都在目标色域之内，记做  $G(C,L)$ ，以减小图像的形态改变。下面将对其中的每一步都进行详细的介绍。

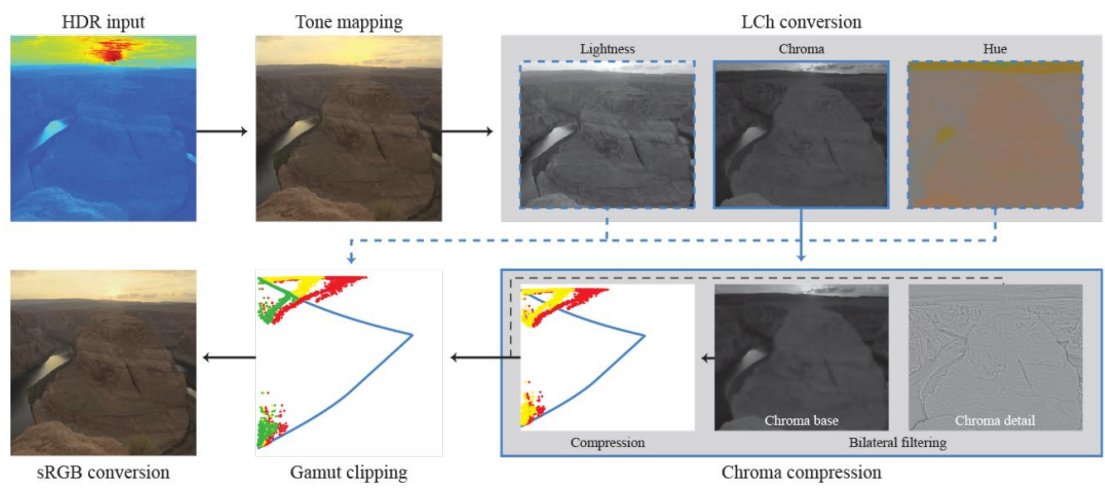


图 8-10 算法流程图

## 亮度压缩

由于算法希望通过色域压缩以及随后的色域截断步骤校正色调映射过程中产生的所有失真，因此，算法的第一步，就是利用色调映射，对亮度进行压缩。任何现存的色调映射算法，理论上都可以直接集成到这个色域映射的框架中，作为第一步亮度压缩中使用的色调映射算法。这里以 Photographic TMO [7] 为例，介绍亮度压缩的整个流程与步骤。

1. 输入的 HDR 图像被转换到 TMO 所需的表示法下。（本例中，转换到 CIE xyY）
2. 利用色调曲线，对通道 Y 进行映射，得到压缩后的亮度信道  $Y_c$ 。这里并不限制色调映射算法的类别，全局算法与局部算法都可以在此处应用。
3. 压缩后的亮度重新添加到图像当中，并且转换为 XYZ 表示法。
4. 最后，色调映射后的图像  $I$  被正则化，使得 Y 的最大值设置为 100，以供后续处理。在 Photographic TMO 的例子中，由于输出的亮度值在 0-1 之间，因此需要进行比例放大，将亮度值进行改变，以供后续处理。

这种亮度压缩的流程，提供了非常高的色调映射的灵活性，使用者可以使用任何想要使用的色调映射算法，但是，这种灵活性的代价是后续需要进行大量的计算来校正色调映射算法带来的色域改变，因为很多色调映射算法在设计的时候并没有考虑色域的问题，仅仅考虑了亮度的压缩。

## 色域边界计算

由于这种算法的设计目的是避免不需要的色度与色相偏移，并且避免不可控的色域截断（即使被截断的色域在目标色域之外），因此，想要实现这些目标，就需要将这些

颜色参数分开。因为如果这些颜色参数或者称之为颜色信息混合在一起的话，调整某一参数时，另一个参数也会被同时改变，这不利于算法的设计与优化。因此，该算法将图像的颜色使用 CIE  $L^*C^*h^*$  表示，也就是 CIE  $L^*a^*b^*$  的圆柱形表示法。这两种表示法在传统的色域映射算法 [1] 中也经常被使用。

为了确定正确的压缩量以及评估一个给定的像素点是否处在输出色域中，算法需要首先确定色域的边界。在这种算法中，目标色域被设置为 sRGB，在转换到 Lab 时使用 D65 白点。同时，目标色域与白点也可以根据需求设置为其他值。输入色域则是输入的 HDR 图像所包括的颜色范围。计算输入与输出色域的边界的方法基于 [1] 提供的获取 sRGB 色域在 LCh 表示法下的色域边界。图 11 显示了输入色域与输出色域的极端差异。在明度与色度上，两个色域的极端差异都在 10 倍以上。

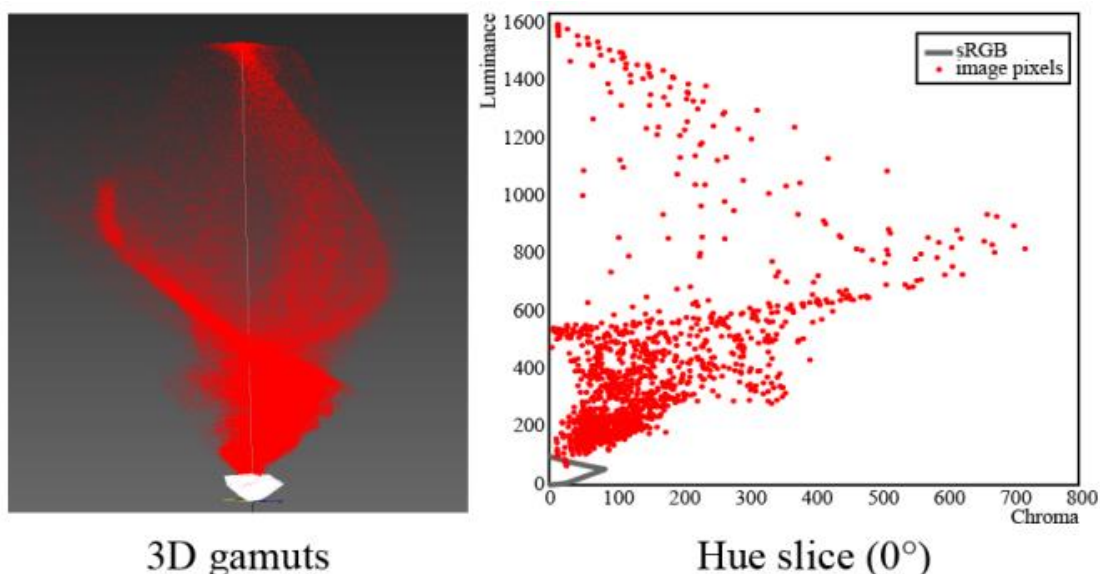


图 8-11 输入与输出色域的极端差异

## 色度压缩

对于 HDR 图像来说，图像中的大量像素点的色度值都在目标色域之外。如果对色域的压缩进行不恰当处理的话，会造成这些像素点的色相发生不好的偏移。并且，亮度域的色调压缩会造成图像的饱和度增大。为了校正这些现象，有两种算法可以被用来压缩色度  $I(C)$ 。

### 色相特性方法

这种方法将色度值  $I(C)$  通过两步进行压缩。第一步是为每个色相  $h \in \{1^\circ \dots 360^\circ\}$  确定一个比例因子  $R_h$ ，这样就可以构成一个向量  $\mathbf{R}$ 。为了实现这一目标，算法通过按比例缩放色域边界  $G_h(C, L)$ ，直到这个色域边界能够完全地包括住所有这一色相  $h$  下的像素点。

接下来正式地定义这种算法：对于一个色相，定义它的初始色相比例  $R_{h,0} = 1$ 。如果存在这一色相下的点在色域范围之外，在每个迭代步骤  $i$  处，增加比例  $R_{h,i}$  并且按比例缩放色域边界：

$$R_{h,i} = R_{h,i-1} + d$$

$$G_{h,i}(C, L) = \begin{bmatrix} R_{h,i} & 0 \\ 0 & R_{h,i} \end{bmatrix} G_h(C, L) \quad (6)$$

这里的增量  $d$  设置为一个较小的量即可（通常设置为 0.1）。

尽管使用这里的比例系数可以直接对源图像进行处理，从而将源图像的色度压缩到目标色域之中，但是在实际操作过程中，如果利用源图像的色域整体来得到  $R$ ，然后进行压缩，这时会产生一种非常差的效果，因为源图像的色域非常大，因此如果  $R$  来自于源图像的整体，由于一些及其明亮的像素点的存在，整幅图像的平均色度会被压缩得过低。因此，在实际应用当中，通常不会利用源图像的整个色域来得到  $R$ ，而是取其中的一部分色域。

在实际操作过程中，选取哪部分色域，以及选取多大的色域，是一个重要的考量。色域选取的比例取决于图像内容分布的。如果整幅图像中都存在需要被压缩的像素点（即存在色度超出目标色域的像素点），那么选取的比例就可以很大。反之，如果只有一小片区域的像素需要被压缩，那么就选取很小的比例，从而避免图像的效果变差。在实际应用过程中，判断需要被压缩的像素是否分布与全图的方式是计算出需要被压缩的像素构成的连通分支的数量，然后再计算出其中包含的像素的数量，两者的比值就是判断的标准。通常如果这个比值小于 0.01，那么选取的比例就不宜过大。

为了保持图像细节的良好，而不会由于很多边缘被平滑操作而产生细节丢失，图像的色度通道  $I(C)$  需要首先通过一个双边滤波器（ $\sigma_s = 0.2 \max(I_{width}, I_{height})$ ,  $\sigma_r = 0.05 \max(I(C))$ ）来获取 base layer,  $I(C)_{base}$ ，因此实际上用于（6）中的  $I(C)$  实际上是  $I(C)_{base}$ 。而获取 detail layer 的方式可以是减法，也可以是除法，两者的实际效果基本一致。算法中实际应用的，是通过除法方式来获取 detail layer:  $I(C)_{detail} = I(C)/I(C)_{base}$ 。

尽管上述的方法已经能够修正大多数的失真，如果直接使用不同色相下的  $R_h$ ，将会在最后的图像中间歇地产生不连续，原因是相邻的色相之间，可能存在着一些细微的改变，而这种改变将会影响最后的结果。因此，在最后的比例放缩之前，需要对  $R$  进行平滑操作，从而得到  $R'$ 。这里使用的平滑函数是 lbox（averaging box）函数，因为这种平滑函数在性能优秀并且运算效率较高。

最终，在每个输出图像的每个色相  $h$  进行比例缩放：

$$I(C)'_{base,h} = \frac{1}{R'} I(C)_{base,h} \quad (7)$$

接着，再使用 detail layer 来重构最终的色域压缩后的图像  $I(C)' = I(C)'_{base} \times I(C)_{detail}$ ，其中， $\times$  表示元素相乘。

### 全局方法

前面描述的色相特性方法可以很好地压缩色度，并且可以最大化地利用色域空间。而这种方法的代价是计算复杂度很高。同时也可以发现，色度通道的动态范围与显示色域比较起来实际上并不是很大（这里的动态范围并不是指 HDR 的动态范围，只是表示色度通道的最大值与最小值的比值）。因此，一个线性的压缩算法可能也能提供不错的效果。因此，在这种前提下，有另一种更简单，低计算复杂度的方法，也可以投入使用。前面提到的色相特性方法，可以用于色调映射的后续处理，因为在这种情况下，精确度是首要的考量；而在其他一些对精确性要求不高，而计算能力有限的情况下，例如显示设备的色域映射，就可以使用这里提到的线性映射方法。

在全局方法中，仍然将像素按照 0-359 的色相角进行分类，对每一类进行压缩。但是与前面方法不同的是，色相特性方法针对每个不同的色相角使用不同的压缩系数，而在全局方法中，为了减少计算量，使用所有色相角中最小的系数，对所有的色相角进行压缩。

类似于色相特性方法，这里的全局压缩方法也只对 base layer 进行压缩：

$$I(C)'_{base,h} = I(C)_{base,h} \min_{h \in [0^\circ, 359^\circ]} \left[ \frac{Cusp_{d,h}}{Cusp_{s,h}} \right] \quad (8)$$

这种方法只在  $\min[Cusp_{d,h}/Cusp_{s,h}] < 1$  时，对色度进行压缩。

此时，全局算法与色相特性算法一样，也会由于运用这个源色域而导致色度压缩的时候产生过度压缩，进而影响图像质量。为了避免这种影响，全局算法中也使用了相同的利用部分源色域的方式来避免过度压缩。但是，全局算法还需要额外进行一步操作，以将部分仍然处在色域边界之外的像素点转换到目标色域内。这种方法将在下一部分进行介绍。

## 色域截断 (Gamut Clipping)

由于算法到目前为止，色度与亮度的运算与处理都是分开的，因此不能保证将两者合并起来之后，颜色不会处在色域边界之外。这种现象可以见图 12。在色调映射过程中，图像的亮度通道的值会被压缩。尽管色调映射后的图像的最大亮度值会被保证为 100nit，但是可能仍然处在色域边界之外（红色点）。利用色度压缩能够有效地减轻这个问题（黄色点），使得像素点向色域边界移动，但是这并没能完全解决这个问题。为了将所有像素点变换到色域边界之内，同时不影响已经处在色域边界中的像素点，算法在最后一步，使用了色域截断处理。由于像素点的色度与亮度可能都在色域边界之外，色域截断处理需要在色度与亮度两个通道上进行，因此在实际处理过程中，需要考虑两方面的因素，并且可能两者的优化不可兼得。

尽管现在有很多算法可以实现色域截断，并且能够给每个色域边界之外的像素点一条移动路径，由此可以将这个像素点移动到色域边界内，但是这些算法的应用场景都是只考虑色度一个方面，而不考虑亮度。在本算法的特殊应用场景里，色域边界之外的点都是高亮度与高色度的点。

由于高亮区域的点难以处理，因此在调整色度与亮度时，需要在两者之间保持微妙的平衡以保证结果图像的外观不会发生很大的改变。这里使用的方法是，在同一个色相角的系列像素中，对于一个像素点  $p \in I'$ ，在色域边界上可以确定一个点  $P_{clipC}$  使得  $P_{clipC}(C) \in G(C)$ ， $P_{clipC}(L) \in p(L)$ 。类似的，再确定一个值  $P_{clipL}$ ， $P_{clipL}(C)$  的值不变，并且  $p(L)_{clipL}$  移动到色域边界上。计算出这两个点后，利用线性插入将色域外的像素映射到目标色域的色域边界上。



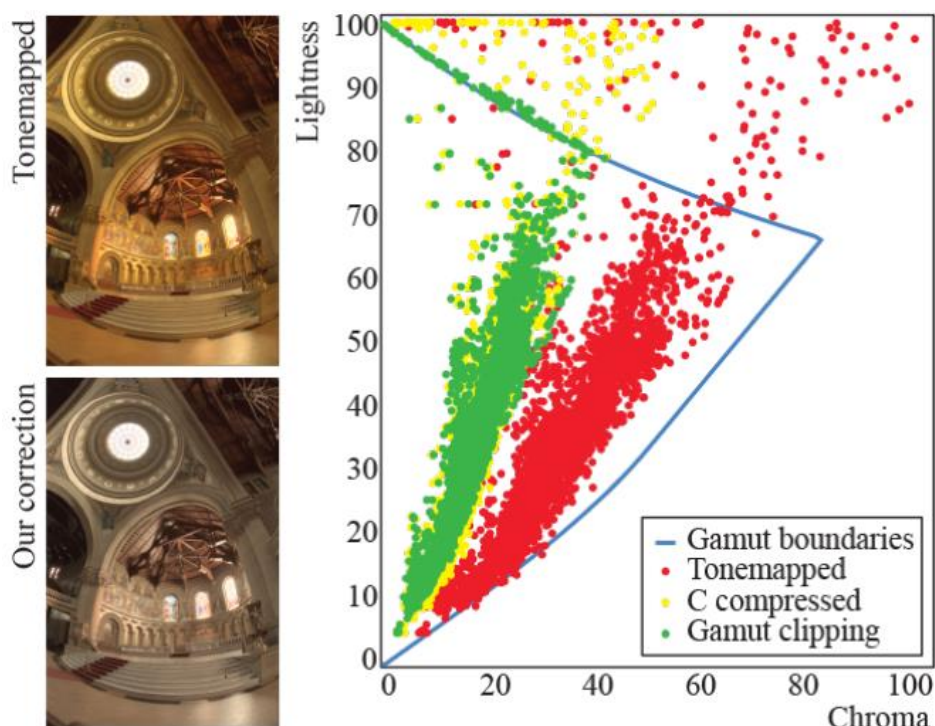


图 8-12 色调映射后，很多点实际上都落在了色域边界之外（如图中红色点所示），接着，随着色域压缩算法的使用，像素点不断地逼近色域边界，但是仍然有一部分像素点处在色域边界之外

## 参考文献

- [1] J. Morovic, Color Gamut Mapping, Barcelona, Spain: Hewlett-Packard Company, 2008.
- [2] CIE (2004d), CIE 156:2004: Guidelines for the evaluation of gamut mapping algorithms, CIE Central Bureau, Vienna, 2004.
- [3] ITU, ITU-R BT.709: basic parameter values for the HDTV standard for the studio and for international programme exchange, in: Standard Recommendation 709, International Telecommunication Union, 1990.
- [4] International Telecommunication Union, Recommendation ITU-R BT.2020 - 2, 2015.
- [5] ITU, Recommendation ITU-R BT.2407-0: Color gamut conversion from Recommendation ITU-R BT.2020 to Recommendation ITU-R BT.709, International Telecommunication Union, 2017.
- [6] ITU, Recommendation ITU-R BT.2087-0: Color conversion from Recommendation ITU-R BT.709 to Recommendation ITU-R BT.2020, International Telecommunication Union, 2015.
- [7] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, Photographic tone reproduction for digital images, in: *ACM Transactions on Graphics*, vol. 21, no. 3, p. 267–276, 2002.
- [8] Fairchild, M. D., hdr-CIELAB and hdr-IPT: Simple Models for Describing the Color of High-Dynamic-Range and Wide-Color-Gamut Images, in: *IS&T/SID, 18th Color and Imaging Conference*, 2010.



- [9] Fairchild, M. D. and Chen, P.-H., Brightness, lightness, and specifying color in high-dynamic-range scenes and images, in: *Proc. SPIE 7867, Image Quality and System Performance VIII* , 2011.
- [10] Elena Sikudova, Tania Pouliy, Alessandro Artusiz, Ahmet Oguz Akyuzx, A Gamut Mapping Framework for Color-Accurate Reproduction of HDR Images, in :IEEE Computer Graphics and Applications, 2016